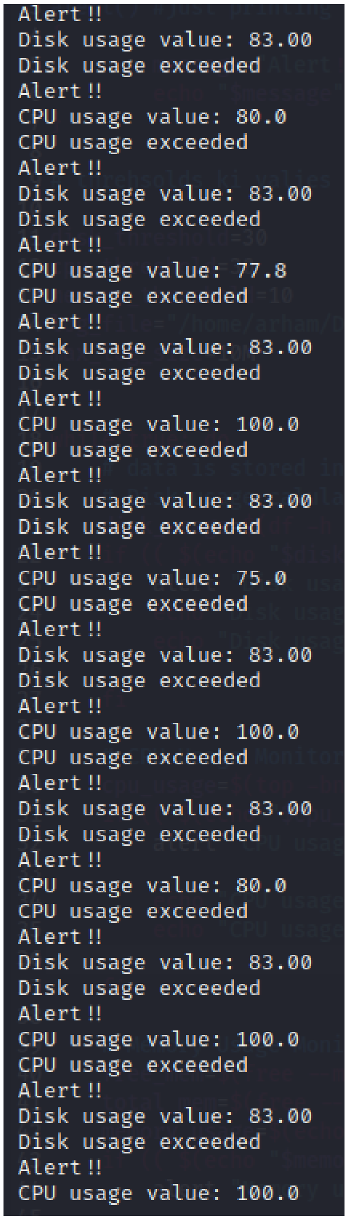
I used chmod +x q1.sh to compile the code and then used bash to execute the code.

Below is shown the output of the code.



```
Alert !!
Disk usage value: 83.00
Disk usage exceeded
Alert !!
CPU usage value: 80.0
CPU usage exceeded
Alert !!
Disk usage value: 83.00
Disk usage exceeded
Alert !!
CPU usage value: 77.8
CPU usage exceeded
Alert !!
Disk usage value: 83.00
Disk usage exceeded
Alert !!
CPU usage value: 100.0
CPU usage exceeded
Alert !!
Disk usage value: 83.00
Disk usage exceeded
Alert !!
CPU usage value: 75.0
CPU usage exceeded
Alert !!
Disk usage value: 83.00
Disk usage exceeded
Alert !!
CPU usage value: 100.0
CPU usage exceeded
Alert !!
Disk usage value: 83.00
Disk usage exceeded
Alert !!
CPU usage value: 80.0
CPU usage exceeded
Alert !!
Disk usage value: 83.00
Disk usage exceeded
Alert !!
CPU usage value: 100.0
CPU usage exceeded
Alert !!
Disk usage value: 83.00
Disk usage exceeded
Alert !!
CPU usage value: 100.0
```

These alerts are also being stored in the Arham.log file.

```
alert() #just printing alert here
{
        message="Alert!!"
        echo "$message" | tee -a "$log_file" # i used tee -a so that the terminal also shows the echo statements and then they are aslo written in the log file named arham.log
}
```

*** the above is the alert function which simply prints alert! On the terminal and writes it down to the log file

```
# setting the threshold values below


disk_threshold=30
cpu_threshold=30
memory_threshold=10
log_file="/home/arham/Desktop/OS thoery/Assignment # 1/arham.log"
max_log_size=10M



while true; do
  # data is stored in the file after each print in terminal
  # Disk usage calculations and then writing alert call
  disk_usage=$(df -h --output=pcent / | sed '1d' | awk '{printf "%.2f", $1}')  #df displays disk usage and used output=pcent to output the % only
  if (( $(echo "$disk_usage > $disk_threshold" | bc -l) )); then # checking the threshold using if , if is doing the same thing throughout the code
    alert "Disk usage exceeds $disk_threshold%: $disk_usage"
        echo "Disk usage value: $disk_usage" | tee -a "$log_file"
    echo "Disk usage exceeded" | tee -a "$log_file"


  fi
```

```
# CPU Usage Monitoring and then calling alert
cpu_usage=$(top -bn1 | grep "Cpu(s)" | sed "s/.*, *\([0-9.]*\)%* id.*/\1/")
if (( $(echo "$cpu_usage > $cpu_threshold" | bc -l) )); then
  alert "CPU usage exceeds $cpu_threshold%: $cpu_usage%"

      echo "CPU usage value: $cpu_usage" | tee -a "$log_file"
  echo "CPU usage exceeded" | tee -a "$log_file"


fi


# Memory Usage Monitoring and then alert is called if it is below threshold, in the above we were checking if it was above threshold
free_mem=$(free --mega | awk '/Mem/{print $4}')  # free mega displays the free memory
total_mem=$(free --mega | awk '/Mem/{print $2}')
memory_usage=$(echo "scale=2; ($total_mem - $free_mem) * 100 / $total_mem" | bc)
if (( $(echo "$memory_usage < $memory_threshold" | bc -l) )); then
  alert "Memory usage falls below $memory_threshold%: $memory_usage%"

      echo "Memory usage value: $memory_usage" | tee -a "$log_file"
  echo "Memory usage exceeded" | tee -a "$log_file"


fi


# Log Rotation means that new file is made when old one is filled up to 10mb
if [ -f "$log_file" ]; then # -f tells us if the path to the file is the one we need or not
  log_size=$(du -b "$log_file" | awk '{print $1}')
  if [ "$log_size" -gt "$(($(echo "$max_log_size" | tr -d '[:alpha:]') * 1024))" ]; then
    mv "$log_file" "$log_file.old"
```

```
            touch "$log_file"
            alert "Log file rotated: $log_file"
        fi
    fi

    sleep 10s
done
```